# MSDA-Text: Template-Guided Long-Form Text Generation with Multi-Source Data Augmentation

Zheng Dai[1], Yilun Zhang[3], Pengjia Wang[1], Qianpu Jiang[1], Fuguo Liu[4], Yufeng Shi[1,2]*
{daiz@mail.sdu.edu.cn, ylzhang@mail.sdu.edu.cn, 202411936@mail.sdu.edu.cn, jiangqianpu@mail.sdu.edu.cn, lfg53880@cjc.edu.cn, yfshi@sdu.edu.cn}
[1] Institute for Financial Studies, Shandong University, Jinan 250100, China
[2] State Key Laboratory of Cryptography and Digital Economy Security, Shandong University, Jinan 250100, China
[3] Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266237, China
[4] School of Mathematics and Data Sciences, Changji University, Changji 831100, China
* Corresponding author

**Abstract.** Large Language Models (LLMs) have demonstrated remarkable capabilities in text generation, yet their outputs often depend heavily on pre-training data and lack the factual depth required for domain-specific long-form writing, such as industrial reports or biographical summaries. To address this limitation, we propose MSDA-Text (Template-Guided Long-Form Text Generation with Multi-Source Data Augmentation), a framework designed to produce accurate and comprehensive long-form texts aligned with user intent. Building upon existing long-text generation architectures such as Storm, MSDA-Text introduces two key enhancements: (1) a template-guided outline generation process that incorporates user-provided reference materials into multi-perspective LLM discussions, and (2) multi-source data augmentation that integrates both Internet-based and local real-time data through Retrieval-Augmented Generation (RAG) and Text-to-SQL techniques. The framework employs the Model Context Protocol (MCP) to unify template parsing across heterogeneous file types and features a long-text writing agent that autonomously retrieves and synthesizes content for each outline section. Experimental results demonstrate that MSDA-Text generates long-form documents that are more structured, user-aligned, and factually grounded than existing LLM-based methods.

**Keywords:** Large Language Models, Long-Form Text Generation, Multi-Source Data Augmentation, Template-Guided Generation, RAG, Text-to-SQL.

## 1 Introduction

Since their emergence, large language models (LLMs) have demonstrated unprecedented capabilities in text generation [1]. However, the content they produce is often limited by their pre-training

IIKI 2025, December 18-21, Chengdu, People's Republic of China

DOI 10.4108/eai.18-12-2025.2365293

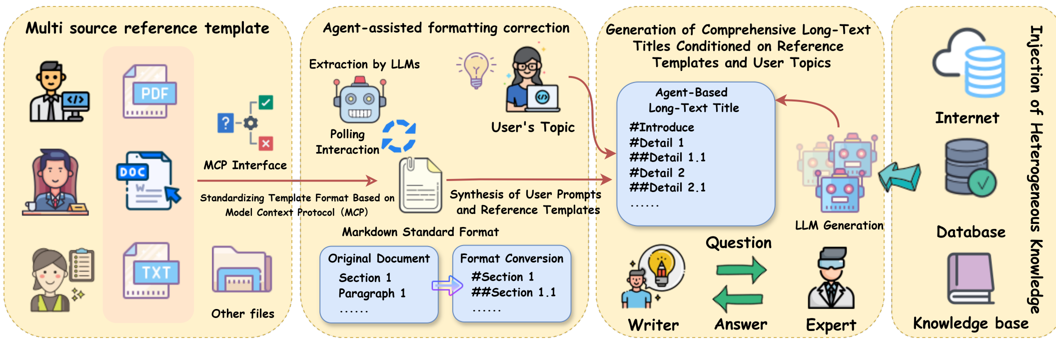


**Fig. 1. Overall Architecture of MSDA-Text**. This figure illustrates the Template-Guided Long-Form Text Generation Strategy with Multi-Source Data Augmentation.

data and fails to meet the demand for well-grounded, domain-specific long-form texts such as industrial reports or biographical summaries. This study aims to construct a long-text generation model that leverages broader data sources while better aligning with user requirements, with the goal of producing comprehensive and accurate industrial reports.

For instance, researchers from Stanford University proposed the Storm framework [2], enabling LLMs to generate well-structured long-form articles such as complete Wikipedia pages. Storm summarizes the process of generating long-form text into two key stages:

- Generating a detailed outline of the article through LLM-based multi-agent discussions, followed by retrieving a set of relevant reference documents based on that outline;
- Using the outline as the core structure to fill in content drawn from the reference materials, thereby producing a coherent and informative long-form article.

The Storm framework highlights the importance of incorporating external resources for enhancing long-form generation quality. However, much domain-specific information cannot be easily obtained through simple topic-based retrieval. To enhance LLMs' research capabilities, Storm employs multi-perspective discussions to synthesize a more comprehensive thematic outline before data collection and content generation.

Building upon this foundation, our work seeks to develop a more refined, precise, and real-time long-text generation system, with two key improvements:

- The outline produced from LLM multi-perspective discussions often diverges from users' expectations. We aim to improve this process by allowing user-provided guidance or hints to influence the discussion and outline generation, ensuring that the resulting structure aligns more closely with user intent.
- While existing frameworks rely primarily on Internet data—which, although superior to traditional outline-driven RAG architectures [3], may lack access to unpublished or proprietary

data—we aim to enable our system to incorporate real-time local data updates into the generated long-form text.

To address the misalignment between LLM-generated outlines and user expectations [4], our approach introduces an interface for user-provided reference templates, allowing users to upload long-text exemplars or related documents. The system analyzes these templates to extract outline structures and latent thematic information. Since uploaded templates can vary in form—from well-structured formal documents to loosely organized text lists—we employ LLM-based tools to adaptively extract the necessary information. To unify the handling of heterogeneous input formats (e.g., plain text, PDF, etc.), our framework adopts the Model Context Protocol (MCP) [5], which standardizes parsing and processing across diverse document types.

To enrich the data sources used for long-text generation, we integrate Retrieval-Augmented Generation (RAG) and Text-to-SQL technologies into our architecture. This enhancement enables the model to query and utilize private databases or organizational knowledge bases in addition to open Internet sources, thereby producing long-form reports that are both up-to-date and contextually relevant. In this way, the generated text can draw upon multi-source data, encompassing both structured (database) and unstructured (knowledge base) content.

Based on the above considerations, we propose **MSDA-Text** (Template-Guided Long-Form **Text** Generation with **M**ulti-**S**ource **D**ata **A**ugmentation), a new architecture for user-guided, data-enriched long-text generation, as illustrated in Fig. 1.

The system extracts outline cues from user-provided templates as guidance, initiates automated LLM discussions to derive the article structure, and connects to multiple data sources— including Internet documents, local databases, and knowledge bases— to populate content for each section of the generated text.

Our main contributions are as follows:

- **Template-Guided Generation Strategy.** We propose the MSDA-Text framework, which introduces a template-guided long-text generation mechanism. It provides an interface for users to upload reference templates and employs MCP-based processing to uniformly extract and parse the uploaded files in different formats.

- **Outline Extraction and Semantic Mining.** MSDA-Text analyzes the uploaded template to extract explicit outline structures or mine implicit semantic cues, thereby enabling outline-constrained generation and targeted material retrieval. The generated outlines more accurately reflect user expectations compared with prior LLM-driven approaches.

- **Long-Text Writing Agent.** We design a long-text writing agent that autonomously retrieves and synthesizes content for each section title based on the generated outline. This agent-driven process enhances the coherence and factual grounding of the resulting long text.

- **Multi-Source Data Integration.** MSDA-Text incorporates multi-source data retrieval methods, ensuring that generated texts no longer depend solely on the Internet or a single data source. Real-time updates from both structured databases and unstructured knowledge repositories can be dynamically integrated into the generation process.

## 2 Related Work

This section provides an overview of existing long-form text generation frameworks and related methods that can facilitate the generation of long texts.

### 2.1 The STORM Architecture

The STORM architecture, proposed by researchers at Stanford University, is a structured framework designed to enhance the long-form generation capabilities of LLMs. It decomposes the generation process into two main stages to ensure that the produced text is coherent, comprehensive, and well-grounded:

- **Outline Generation through Multi-Perspective Discussion:** In this stage, the system uses LLMs to simulate a multi-agent discussion, exploring the target topic from different perspectives [6]. Through this interaction, a detailed and logically structured outline is produced, capturing the key aspects and organizational structure of the intended article.

- **Content Completion with Retrieved References:** Once the outline is established, the system retrieves a set of reference documents relevant to each section of the outline. These references serve as the factual basis for content generation. The LLM then fills in the outline by generating text based on the retrieved information, resulting in a long-form document that is both information-rich and contextually accurate.

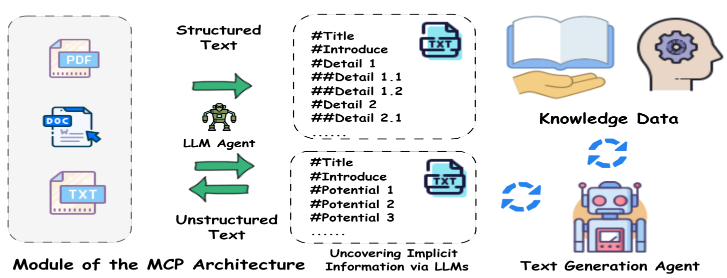


**Fig. 2.** Template-driven outline extraction and recognition architecture based on MCP.

STORM emphasizes the importance of external knowledge in the long-form generation process. By decoupling structural planning from actual content generation, it achieves greater factual accuracy and thematic coherence. However, its reliance on publicly available internet data introduces certain limitations when generating documents based on up-to-date or private data sources.

### 2.2 Model Context Protocol

The Model Context Protocol (MCP) is a unified processing interface designed to standardize the ingestion, interpretation, and contextualization of different input formats. It enables the model to effectively handle a variety of document types, including plain text files (.txt), structured documents (e.g., Markdown), and semi-structured formats such as PDF or Word files.

MCP parses the normalized content to extract key structural elements, such as headings, sections, lists, or bullet points, and maps them to a semantic outline that can be interpreted by the language model. For unstructured inputs, it employs LLM-based heuristic methods to infer latent structures or implied intent.

### 2.3 Retrieval-Augmented Generation Model

The Retrieval-Augmented Generation (RAG) model [7] retrieves relevant information from a given data source and uses it to generate the desired output. Researchers primarily proposed this concept for retrieving unstructured knowledge. For many LLM systems, external data is often necessary to achieve superior text generation. However, in most cases, the external data may not be structured in a way that allows for quick retrieval, such as data stored in databases or spreadsheets. Instead, this information is typically stored in text files, like notepad documents. As such, the RAG model is capable of retrieving relevant textual data as needed, enabling the LLM system to generate more accurate and user-aligned text [8].

### 2.4 Text-to-SQL

Text-to-SQL is a task in natural language processing that focuses on converting natural language queries into executable SQL statements [9]. It allows users to interact with structured databases using simple language, thereby bridging the gap between non-technical users and complex data querying systems.

Similar to RAG, the primary purpose of Text-to-SQL is to provide private data to LLM systems for reference, helping to generate the desired text. However, unlike RAG, the primary use of this technology is to retrieve structured data stored in databases. This type of data is typically well-organized, large-scale, and detailed, and by retrieving this precise information, Text-to-SQL serves as a reference for the LLM.

In the context of long-form text generation, incorporating Text-to-SQL allows the system to dynamically retrieve structured data from relational databases based on user intent [10]. This ensures that the generated content is not only relevant and informative but also grounded in real-time, accurate data. By leveraging Text-to-SQL, the framework proposed in this study seamlessly integrates private or domain-specific structured data sources into the generation process, enhancing both precision and reliability.

## 3 Proposed Method

This section introduces the overall design of the proposed **MSDA-Text** framework, which integrates template-guided outline generation, multi-source data augmentation, and a controllable long-text generation process.

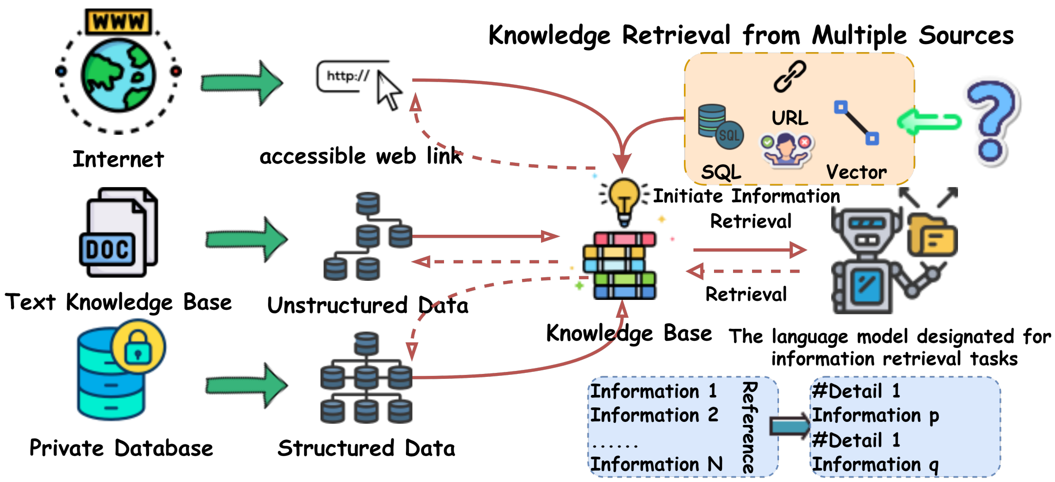


**Fig. 3.** Multi-source knowledge acquisition and fusion architecture.

### 3.1 Notation

For clarity, we define the main symbols used throughout this section:

- $\mathscr{T} = \{t_0, t_1, \ldots, t_n\}$ : Titles extracted from user-uploaded `.txt` templates.
- $\mathscr{W} = \{w_0, w_1, \ldots, w_m\}$ : Titles extracted from `.docx` templates.
- $\mathscr{P} = \{p_0, p_1, \ldots, p_k\}$ : Titles extracted from `.pdf` templates.
- $\mathscr{C}$ : Combined reference title set used for outline construction.
- $S_d$ : Source reliability score for data source $d$.
- $\lambda_d$ : Weight assigned to source $d$ when fusing retrieved information.
- $\mathscr{D} = \{D_{\text{web}}, D_{\text{local}}, D_{\text{kb}}\}$ : Set of all available data sources (web, local database, and knowledge base).

### 3.2 Template-Guided Architecture Based on MCP

To ensure that MSDA-Text generates article outlines that accurately reflect user intent, users are allowed to provide custom templates at the outline generation stage. Based on the extracted outlines and content, the **MCP** constructs a context-aware prompt $\mathscr{P}_{ctx}$ that encodes both user intent and document semantics:

$$\mathscr{P}_{ctx} = f_{\text{MCP}}(U, \mathscr{C}), \tag{1}$$

where $U$ represents user instructions and $\mathscr{C}$ denotes the merged reference set derived as:

$$\mathscr{C} = \mathrm{Sort}\big((\mathscr{T} \cup \mathscr{W} \cup \mathscr{P})\big). \tag{2}$$

This prompt serves as the initialization signal for subsequent generation tasks, enhancing controllability and personalization in long-form text generation. The overall workflow is illustrated in Fig. 2.

### 3.3 Long-Form Text Generation Process

The **MSDA-Text** generation pipeline consists of four major modules: *template parsing*, *outline generation*, *article composition*, and *text refinement*.

#### 3.3.1 Template Parsing Module

The MCP processes user-submitted templates and converts them into structured Markdown format, forming a scaffold for subsequent content generation.

#### 3.3.2 Outline Generation Module

Given a user-defined topic $\tau$ and parsed template $\mathscr{P}_{ctx}$, the LLM performs multi-perspective reasoning to produce a hierarchical outline:

$$\mathscr{O} = f_{\mathrm{LLM}}(\tau, \mathscr{P}_{ctx}), \tag{3}$$

where $\mathscr{O}$ represents the outline structure used to guide document composition.

#### 3.3.3 Article Composition Module

A writing agent retrieves relevant data from heterogeneous sources $\mathscr{D} = \{D_{\mathrm{web}}, D_{\mathrm{local}}, D_{\mathrm{kb}}\}$. For each section $o_i \in \mathscr{O}$, the agent gathers candidate content $\{x_d^i\}$ from each source $d$ and fuses them through a weighted reliability function:

$$C_i = \sum_{d \in \mathscr{D}} \lambda_d S_d x_d^i, \quad \text{where } \sum_d \lambda_d = 1. \tag{4}$$

This mechanism ensures factual precision by prioritizing structured local data ($S_{\mathrm{local}} > S_{\mathrm{web}}$).

#### 3.3.4 Text Refinement Module

Post-processing operations remove redundancy and correct stylistic inconsistencies. The final text $T_{\mathrm{final}}$ is obtained as:

$$T_{\mathrm{final}} = f_{\mathrm{refine}}(C_1, C_2, \ldots, C_n). \tag{5}$$

### 3.4 Multi-Source Data Fusion and Retrieval

To enrich data sources beyond the Internet, MSDA-Text integrates local databases and internal knowledge bases. These data are categorized as structured or unstructured, and handled respectively by **Text-to-SQL** and **RAG** modules (see Fig. 3).

#### 3.4.1 RAG Model

For unstructured text retrieval, the LLM automatically formulates semantic queries $q_t$ based on task intent:

$$q_t = f_{\text{prompt}}(\text{intent}), \tag{6}$$

and retrieves relevant passages using a retriever $\mathscr{R}$:

$$R_t = \mathscr{R}(q_t, D_{\text{kb}}). \tag{7}$$

This design allows context-aware information augmentation during content generation.

#### 3.4.2 Text-to-SQL Module

Structured data are accessed using a text-to-SQL strategy [11, 12, 13]:

$$\text{SQL}_t = f_{\text{LLM}}(q_t, \mathscr{S}), \tag{8}$$

where $\mathscr{S}$ denotes the schema or creation statements of the database tables. The generated SQL query is executed to obtain structured results [14, 15, 16]:

$$R_{\text{SQL}} = \text{Exec}(\text{SQL}_t, D_{\text{local}}). \tag{9}$$

Normalization procedures are applied to ensure consistent query forms, as illustrated in Fig. 4.

Through these modules, MSDA-Text effectively combines template-guided reasoning, user-aligned control, and multi-source knowledge retrieval to produce comprehensive, accurate, and up-to-date long-form documents.

Additionally, the normalization of queries is standardized as shown in Fig. 5.

## 4 Experiments

In this section, extensive experiments are conducted to verify the superior performance of the proposed MSDA-Text framework. First, a series of ablation studies are carried out to evaluate the effectiveness of each key component in MSDA-Text, including the template-guided generation and multi-source knowledge retrieval modules. These components are comprehensively analyzed using metrics such as Entity Recall, Soft Recall on Titles, and Template Recall. Then, a human evaluation is performed on the generated report content, where domain experts assess the text quality in terms of Relevance and Coverage. Finally, several visualization results related to the MSDA-Text architecture are presented to further demonstrate its interpretability and effectiveness.

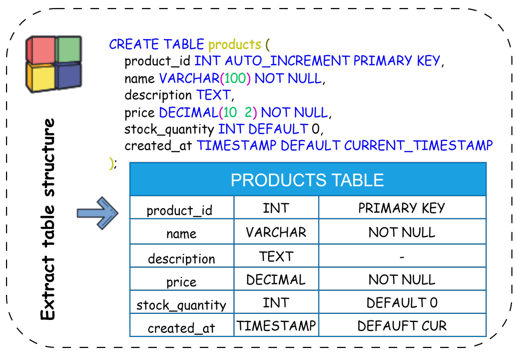


**Fig. 4.** Sample system prompt for Figure 4.

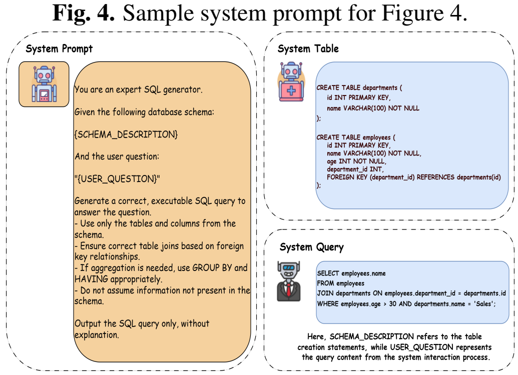


**Fig. 5.** Sample system prompt for Figure 5.

### 4.1 Dataset

In terms of dataset selection, all public data used in this study were obtained from the Internet. For private data, the structured data stored in databases were provided by third parties during a collaborative project with an enterprise from December 2023 to December 2025. Regarding the unstructured text data used to construct the knowledge base, we collected official documents during the project period, primarily sourced from authoritative public datasets such as statistical yearbooks, with a focus on regions like Linyi City. The private dataset mainly covers the current status of regional industrial development. In this work, the quality of long-text generation regarding the industrial development status is adopted as the primary evaluation metric. All private data used in this study were integrated to construct the MSDA-dataset, which primarily includes two knowledge bases obtained from public data repositories and thirteen tables containing information related to industrial development.

### 4.2 Experimental Metrics

In this study, both objective and subjective evaluation metrics are employed to assess the performance of the MSDA-Text framework. For the objective evaluation, we adopt the ROUGE metric suite [17], using Entity Recall to measure the completeness of entity retrieval, with the absolute number of correctly retrieved entities serving as the primary metric. In addition, two percentage-based indicators are introduced: Template Recall and Soft Title Relevance, which respectively measure the degree of adherence between the generated text and the structural template, and the semantic alignment between the generated titles and the expected ones. Note that, for clarity, the percentage symbols are omitted in the reported experimental results.

For the subjective evaluation, ten domain experts with backgrounds in knowledge-base construction were invited to assess the quality of the generated long-form texts. Each expert provided their own input topics and templates, based on which MSDA-Text generated corresponding domain-specific reports. The experts then rated the outputs on a 10-point scale according to content quality and consistency with expectations. The final subjective score was obtained by averaging all expert ratings.

### 4.3 Baselines

To ensure a standardized experimental process and enhance the credibility of the results, we conducted comparative experiments using the same large pre-trained language models. For this purpose, four LLM-based baselines were established based on different configurations of the MSDA-Text framework:

#### 4.3.1 TG-LLM(Template-Guided LLM )

The extracted title and structural information from the user-provided template are directly fed into the LLM for report generation, without integrating any external data sources.

### 4.3.2 SK-MSDA(Single-Source Knowledge MSDA-Text )

The MSDA-Text framework operates with a single-source knowledge setup, where data retrieval is limited to Internet searches only.

### 4.3.3 WT-MSDA(MSDA-Text without Template but with Multi-Source Knowledge)

The MSDA-Text framework enables multi-source data retrieval, including both online and local knowledge bases, but removes the template-guided generation mechanism.

### 4.3.4 MSDA-Text

The complete MSDA-Text architecture is employed, integrating both the template-guided strategy and multi-source knowledge retrieval for comprehensive evaluation.

**Table 1:** Objective Evaluation Results of MSDA Across Different Large Language Models

| Evaluation Metrics | Model | T-G-LLM | SK-MSDA | WT-MSDA | MSDA |
|---|---|---|---|---|---|
| Entity Recall | Deepseek-r1 | 0.79 | 1.04 | 0.75 | **3.26** |
| | Qwen3 | 0.82 | 1.08 | 0.78 | 2.95 |
| | Llama3.1 | 0.85 | 1.10 | 0.81 | <u>3.21</u> |
| | Gemma3 | 0.76 | 0.98 | 0.69 | 3.00 |
| Soft Recall on Titles | Deepseek-r1 | 11.49 | 25.07 | 40.50 | **50.14** |
| | Qwen3 | 12.10 | 26.37 | 41.35 | <u>50.00</u> |
| | Llama3.1 | 11.80 | 27.95 | 39.70 | <u>50.00</u> |
| | Gemma3 | 10.70 | 23.50 | 33.90 | 49.85 |
| Template Recall | Deepseek-r1 | 15.83 | 45.17 | 58.33 | <u>94.12</u> |
| | Qwen3 | 17.30 | 46.55 | 59.33 | 93.50 |
| | Llama3.1 | 16.50 | 43.33 | 62.15 | **95.02** |
| | Gemma3 | 14.80 | 43.09 | 55.73 | 89.48 |

*Note:* This table presents the objective evaluation results of the proposed MSDA-Text framework compared with baseline methods (T-G-LLM, SK-MSDA, WT-MSDA) across different large language models. The evaluated models include Deepseek-r1, Qwen3, Llama3.1, and Gemma3: Deepseek-r1 is a reasoning-focused model with a reinforcement learning-driven architecture that does not rely on supervised fine-tuning, and it includes the DeepSeek-R1-Zero variant; Qwen3 is the latest version of Alibaba's Qwen series, characterized by a 32768-token context window and "Thinking Mode" to enhance reasoning capabilities; Llama3.1 is an open-weight model developed by Meta Platforms, Inc., available in 8B, 70B, and 405B parameter versions, and optimized for instruction following and coding tasks; Gemma3 is a lightweight and efficient open language model from the Gemma Team (affiliated with Google), with strong performance across diverse task types. The evaluation metrics include **Entity Recall** (used to evaluate the effectiveness of entity extraction), **Soft Recall on Titles** (used to assess the accuracy of title alignment), and **Template Recall** (used to measure the degree of adherence to structural templates). In the table, the best results are shown in **bold**, and the second-best results are <u>underlined</u>.

**Table 2:** Subjective Evaluation Results of MSDA

| Method | Relevance | Coverage |
|---|---|---|
| T-G-LLM | 4.41 | 3.02 |
| SK-MSDA | 4.79 | <u>4.93</u> |
| WT-MSDA | <u>5.11</u> | 3.86 |
| MSDA | **5.45** | **5.07** |

*Note:* This table reports the subjective evaluation of MSDA-Text based on human expert ratings. Ten domain experts scored the generated reports on two criteria: **Relevance** (topic alignment) and **Coverage** (content completeness). Scores are averaged across all participants. The best results are in **bold**, and the second-best are <u>underlined</u>.

### 4.4 Experiment Implementation

This paper conducts ablation studies on each module of the MSDA-Text architecture to evaluate its effectiveness in long-form text generation. In the experiments, we employed multiple large language models, including Qwen and LLaMA, to compute objective metrics such as Entity Recall, Template Recall, and Title Soft Relevance. These results served as the foundation for assessing the objective evaluation criteria.

For the subjective evaluation, the Qwen model was uniformly adopted to generate the long-form content. Ten domain experts were invited to evaluate the generated texts according to two key criteria: Relevance and Coverage [18].

The evaluation results based on objective metrics are presented in Table 1. According to the results, although the evaluation scores vary slightly across different large language models, the MSDA-Text framework consistently demonstrates significant improvements in all related metrics compared with the three baseline models. These improvements highlight the effectiveness of integrating multi-source data and template-guided prompting in long-form text generation tasks.

The template-guided architecture of MSDA-Text, enhanced with multi-source data integration, achieves superior performance—particularly in tasks such as generating comprehensive, domain-specific industry reports. Its ability to incorporate heterogeneous data sources, including real-time updates and expert domain knowledge, substantially improves both accuracy and relevance, making MSDA-Text a promising framework for high-quality report generation.

Furthermore, the subjective evaluation scores assigned by domain experts across the four evaluation metrics are summarized in Table 2. As shown in the results, evaluators consistently awarded higher scores to MSDA-Text in terms of both Relevance and Content Coverage, validating its effectiveness and practical value in long-form text generation.

### 4.5 Experimental Results

This study conducts comparative experiments across baseline models with different structural designs and knowledge configurations to evaluate the effectiveness of the proposed MSDA-Text framework in integrating multi-source data and guiding structured generation. The experimental results demonstrate that the full MSDA-Text architecture consistently outperforms all baseline methods across three key metrics: Soft Recall on Titles, Entity Recall, and Template Recall.

Specifically, MSDA-Text achieves the highest score in Soft Recall on Titles, indicating its superior capability to model and leverage title structures during text generation. In terms of Entity Recall, MSDA-Text also shows a notable advantage, suggesting that the integration of heterogeneous knowledge sources leads to more comprehensive coverage of entity-level content. Furthermore, the structure-guided generation mechanism within MSDA-Text significantly enhances Template Recall, reflecting its strong alignment with task-specific formatting and structural requirements.

By contrast, T-LLM leverages structural templates but lacks external knowledge integration, limiting its performance in entity coverage. SK-MSDA, which relies solely on a single knowledge source, improves entity recall but fails to provide structural guidance. WT-MSDA incorporates multi-source knowledge yet omits structural template guidance, resulting in suboptimal overall performance.

In summary, these findings validate the synergistic advantage of the MSDA-Text framework in combining structural understanding with multi-source knowledge fusion. This highlights its potential as an effective modeling paradigm for structured long-form text generation and knowledge-grounded question answering. The visual comparison of the results is presented in Fig. 6.

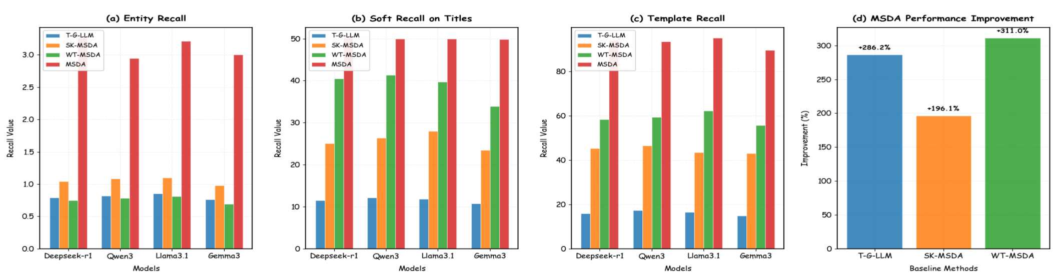


**Fig. 6.** Objective Evaluation Results of MSDA Across Different LLMs.

## 5 Conclusion

The MSDA-Text framework proposed in this study addresses key challenges faced by existing large language models (LLMs) in generating long-form content, particularly in professional fields. These challenges include user alignment, structural guidance, and data reliability. Experimental results demonstrate that MSDA-Text significantly enhances long-text generation quality across three critical dimensions: user alignment, structural guidance, and data consistency.

First, MSDA-Text improves user satisfaction and content relevance by allowing users to upload reference documents that reflect the desired structure and theme. This includes both structured templates and unstructured text. Compared to traditional methods, MSDA-Text enhances parsing and structural inference through the Model-Contextual Prompt (MCP), supporting various input formats (e.g., PDF, DOCX, and plain text). This significantly improves the structural consistency of the generated content.

Second, MSDA-Text integrates Retrieval-Augmented Generation (RAG) and Text-to-SQL techniques to enable the dynamic fusion of both public and private knowledge sources, such as enterprise databases and internal knowledge repositories. This multi-source architecture improves the factual consistency of the generated content and enables the LLM to produce high-fidelity content in real time, based on the given context. Particularly in domain-specific, customized scenarios, MSDA-Text provides a notable advantage.

Finally, both quantitative and qualitative analyses from the experiments show that MSDA-Text outperforms existing baseline models in terms of content relevance, structural consistency, and user alignment. These innovations position MSDA-Text as a scalable and feasible solution for enterprise-level, content-intensive applications, such as report generation, knowledge documentation, and technical writing.

In conclusion, the MSDA-Text framework provides an effective solution to the challenges of long-text generation in professional fields. Its ability to offer structured guidance and integrate multi-source data augmentation leads to significant performance improvements in practical applications, particularly in scenarios that require high levels of customization.

## Acknowledgment

This work was supported by the National Key R&D Program of China (Grant No. 2023YFA1008903), the Major Fundamental Research Project of Shandong Province of China (Grant No. ZR2023ZD33), the Taishan Scholar Project of Shandong Province of China (Grant No. tstp20240803), the Natural Science Foundation of Shandong Province (Grant No. ZR2023LLZ012), and the Key Project of Shandong Provincial Key R&D Program (Soft Science) (Grant No. 2024RZB0204).

## Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o in order to: Grammar and spelling check.

## References


[1] Chang Y, et al. A Survey on Evaluation of Large Language Models. ACM Trans Intell Syst Technol. 2024;15(3):1-45.

[2] Shao Y, et al. Assisting in Writing Wikipedia-like Articles from Scratch with Large Language Models. arXiv preprint arXiv:240214207. 2024.

[3] Gumaan E. ExpertRAG: Efficient RAG with Mixture of Experts–Optimizing Context Retrieval for Adaptive LLM Responses. arXiv preprint arXiv:250408744. 2025.

[4] Shi L, et al. Argumentative Experience: Reducing Confirmation Bias on Controversial Issues through LLM-Generated Multi-Persona Debates. arXiv preprint arXiv:241204629. 2024.

[5] Hou X, et al. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. arXiv preprint arXiv:250323278. 2025.

[6] Hong S, et al. MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. arXiv preprint arXiv:230800352. 2023.

[7] Gupta S, Ranjan R, Singh SN. A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape, and Future Directions. arXiv preprint arXiv:241012837. 2024.

[8] Gao Y, et al. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv preprint arXiv:231210997. 2023.

[9] Katsogiannis-Meimarakis G, Koutrika G. A Survey on Deep Learning Approaches for Text-to-SQL. The VLDB Journal. 2023;32(4):905-36.

[10] Deng N, Chen Y, Zhang Y. Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect. arXiv preprint arXiv:220810099. 2022.

[11] Yi J, Chen G, Shen Z. RH-SQL: Refined Schema and Hardness Prompt for Text-to-SQL. In: Proc. 2024 6th Int. Conf. Electron. Eng. Informatics (EEI). IEEE; 2024. .

[12] Li Z, Wang X, Zhao J, et al. PET-SQL: A Prompt-Enhanced Two-Round Refinement of Text-to-SQL with Cross-Consistency. arXiv preprint arXiv:240309732. 2024.

[13] Shi L, et al. A Survey on Employing Large Language Models for Text-to-SQL Tasks. arXiv preprint arXiv:240715186. 2024.

[14] Chang S, Fosler-Lussier E. How to Prompt LLMs for Text-to-SQL: A Study in Zero-Shot, Single-Domain, and Cross-Domain Settings. arXiv preprint arXiv:230511853. 2023.

[15] Nan L, et al. Enhancing Text-to-SQL Capabilities of Large Language Models: A Study on Prompt Design Strategies. Findings of the Assoc Comput Linguistics: EMNLP 2023. 2023.

[16] Gao D, Wang H, Li Y, et al. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. arXiv preprint arXiv:230815363. 2023.

[17] Lin CY. ROUGE: A Package for Automatic Evaluation of Summaries. In: Text Summarization Branches Out; 2004. .

[18] Kim S, et al. Prometheus: Inducing Fine-Grained Evaluation Capability in Language Models. In: Proc. 12th Int. Conf. Learn. Representations (ICLR 2023); 2023. .